

Table based KNN and AHC Algorithm for Combination of Text Summarization with Text Clustering

Taeho Jo
President
Alpha AI Research
Cheongju, South Korea
tjo018@naver.com

Abstract—In this research, we propose the table based KNN algorithm and the table based AHC algorithm for automating the text summarization with its combination with the text clustering. In the previous work, even if the KNN version was applied to the text summarization, a domain should be presented as an input text tag, manually. In this research, text subgroups are constructed based on their contents by the text clustering, and a novice text is summarized by selecting a classifier which corresponds to its most similar cluster. The table based AHC algorithm, and the table based KNN algorithm are adopted respectively as the approaches to the text clustering and the text summarization. The goals of this research are to automate the text summarization and to improve the performance of both tasks, using the table-based algorithms.

I. INTRODUCTION

The text summarization is defined as the process of presenting the essential part from a text as its summary. It is necessary to decide whether a current text partition belong to summary, or not. The text partition is set as a paragraph, and the text summarization is mapped into the classification of each paragraph into summary or non-summary. The process of text summarization is to partition an input text into paragraphs, classify each paragraph into one of the two categories, and extract paragraphs which are classified with summary. This research focuses on the application of the proposed machine learning algorithm to the classification task which is mapped from the text summarization.

This research is motivated by the necessity of deciding domains manually for designing the text summarization system. The text summarization is viewed into a binary classification, to apply a machine learning algorithm; the text summarization system is decomposed with independent binary classifiers as many as domains. The process of designing the text summarization system is to predefine domains depending on prior knowledge or subjective views, collect sample paragraphs domain by domain, and allocate a binary classifier for each domain. Another motivation of this research is the recommendation of the table based KNN algorithm and the table based AHC algorithm as the better approaches to the text summarization and the text clustering. This research is intended to automate the domain

predefinition by connecting the text summarization with the text clustering.

In this research, we propose the table based KNN algorithm and the table based AHC algorithm as the approaches to the text summarization and the text clustering which relate to each other. In this research, we prepare the text collection where each text is associated with its own summary. The similarity metric between tables is defined, and the KNN algorithm and the AHC algorithm are modified based on the similarity metric into the table-based versions. The collection is segmented into clusters by the text clustering, and a binary classifier which performs the text summarization is implemented for each cluster independently of other clusters. A domain is decided to an input text by its similarities with the cluster prototypes for selecting a corresponding classifier.

Let us mention some benefits from this research. The main problems in encoding texts into numerical vectors are avoided by encoding them into tables. The performances in the text summarization and the text clustering are improved by adopting the table-based versions of the KNN algorithm and the AHC algorithm as the approaches to both tasks. We are free from defining domains depending on prior knowledge and subjective views by automating the domain predefinition through the text clustering. Whenever an input text is given for summarizing a text, we are free from presenting a domain additionally by deciding automatically it based on the similarities of an input text with the cluster prototypes.

Let us mention some remaining tasks as the further discussions on this research. The more advanced machine learning algorithms and deep learning algorithms are modified into the table-based versions. It is necessary to define and characterize mathematically additional operations on tables. The text summarization system which relates to the text clustering should be implemented as a real program or a library. The text classification and the text clustering should relate with the text summarization for improving their performances.

II. PROPOSED WORKS

A. Encoding Process

This section is concerned with the table as the text representation. It is defined as a set of entries, each of which consists of an element and its weight. The table which represents a text consists of entries, each of which is a pair of a word and its weight in the text. The similarity between tables is computed based on shared words as the semantic similarity between texts. This section is intended to describe the process of encoding a text into a table and the process of computing the similarity between tables.

The process of encoding a text into a table is illustrated in Figure 1. It is indexed into a list of words. The weights of words in the text are computed in the text-word weighting; the word weight in the text is a relationship between a word and a text. The entries with their lower weights are removed for downsizing the table. Refer to [1] for studying the entire process in more detail.

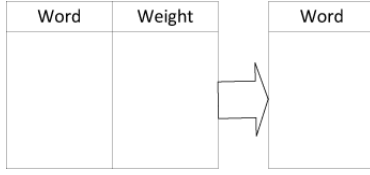


Figure 1. Process of Encoding Texts into Tables

Let us mention the function of a table which maps it into a set of words as shown in Figure 2. The table which represents a text is expressed as entry set as shown in equation (1),

$$T = \{(t_1, w_1), (t_2, w_2), \dots, (t_{|T|}, w_{|T|})\} \quad (1)$$

where t_i is a word, and w_i is the weight of the word, t_i , in the text. The function for generating a list of words is expressed as equation (2),

$$F(T) = \{t_1, t_2, \dots, t_{|T|}\} \quad (2)$$

The function is the operation which takes only words without their weights as a set. The operation is applied to computing the similarity between tables which represent texts.

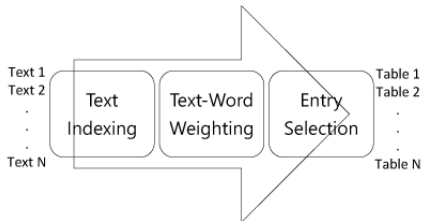


Figure 2. Conversion of Table into Word Set

Let us mention the process of computing the similarity between two tables as the similarity between two texts.

The tables which represent texts are notated by the two sets: $D_1 = \{(t_{11}, w_{11}), (t_{12}, w_{12}), \dots, (t_{1|D_1|}, w_{1|D_1|})\}$ and $D_2 = \{(t_{21}, w_{21}), (t_{22}, w_{22}), \dots, (t_{2|D_2|}, w_{2|D_2|})\}$. The intersection between the two sets which are generated by applying the function, $F(\cdot)$, by equation (3),

$$F(D_1) \cap F(D_2) = \{t_{s1}, t_{s2}, \dots, t_{s|F(D_1) \cap F(D_2)|}\} \quad (3)$$

and the table with entries each of which consists of a shared word, its weight from the table, D_1 , its weight from the table, D_2 , is constructed based on equation (3), as expressed in equation (4),

$$SD_{12} = \{(t_{s1}, w_{s1}^1, w_{s1}^2), (t_{s2}, w_{s2}^1, w_{s2}^2), \dots, (t_{s|F(D_1) \cap F(D_2)|}, w_{s|F(D_1) \cap F(D_2)|}^1, w_{s|F(D_1) \cap F(D_2)|}^2)\} \quad (4)$$

The similarity between the two tables, D_1 and D_2 , is computed by equation (5),

$$sim(D_1, D_2) = \frac{2(\sum_{i=1}^{|F(D_1) \cap F(D_2)|} w_{si}^1 + \sum_{i=1}^{|F(D_1) \cap F(D_2)|} w_{si}^2)}{\sum_{i=1}^{|D_1|} w_{1i} + \sum_{i=1}^{|D_2|} w_{2i}}. \quad (5)$$

The value which is computed by equation (5), is always given as a normalized value between zero and one, as shown in equation (6),

$$0 \leq sim(D_1, D_2) \leq 1. \quad (6)$$

Let us make some remarks on the encoding process and the computation of the similarity between texts. A text is encoded into a table by the text indexing, the word weighting, and the entry selection. A table is expressed as a set of entries, and a table is filtered into a set of words by the defined function. The similarity between tables is computed based on shared words by equation (5). In the future research, we consider representing a text into multiple tables.

B. Table based KNN Algorithm

This section is concerned with the table based KNN algorithm. It was initially proposed as the approach to the text classification by Jo in 2015 [2]. The similarity metric between tables which was described in the previous section is used for computing the similarity between a novice table and a training example. The labels of its nearest neighbors are voted with their identical weights for deciding the label of a novice input. This section is intended to describe the initial version of KNN algorithm from which its variants are derived.

The process of computing the similarity between a novice item and a training example is illustrated in Figure 3. The labeled words which are gathered as samples are encoded into tables, and they are notated by a set $Tr = \{T_1, T_2, \dots, T_N\}$. A novice word is encoded into a table

notated by T . Its similarities with the tables which represent the sample words are computed by equation (??), as $sim(T, T_1), sim(T, T_2), \dots, sim(T, T_N)$. Some training examples which are most similar as the novice input are selected as its nearest neighbors.

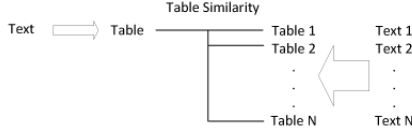


Figure 3. Similarities of Novice Input with Training Examples

In Figure 4, the process of selecting nearest neighbors by the ranking selection is illustrated. The training examples are sorted by the descending order of their similarities, after computing their similarities with a novice example. The training examples with their higher similarities with a novice example are selected as its nearest neighbors, as expressed in equation (7),

$$Nr = Select_k(Tr, T), Nr \subseteq Tr \quad (7)$$

The set of nearest neighbors, Nr , is composed with elements as expressed in equation (8),

$$Nr = \{T_1^{near}, T_2^{near}, \dots, T_k^{near}\} \quad (8)$$

The elements in the set, Nr , are used for voting their labels in the next step.

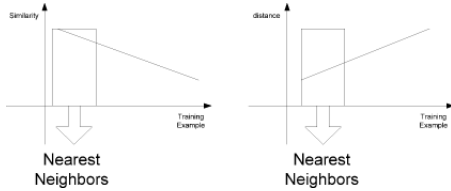


Figure 4. Selection of Most Similar or Least Distant Training Examples as Nearest Neighbors

The process of voting the labels of nearest neighbors for decoding the label of a novice word is illustrated in Figure 5. The predefined categories are notated by $c_1, c_2, \dots, c_m$, and the label of a nearest neighbor is notated by $c_j = label(T_i^{near})$. The number of nearest neighbors which belong to the category, c_j , is notated by $Count(Nr, c_j)$. The label of the novice item, T , is decided by equation (9),

$$label(T) = \arg\max_{j=1}^m Count(Nr, c_j) \quad (9)$$

The process of deciding the label of a novice item by equation (9), is called voting.

Let us make some remarks on the table based KNN algorithm. It computes the similarities of a novice item with the training examples. The training examples with their highest similarities with the novice item are selected as its

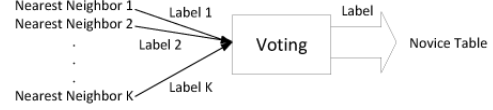


Figure 5. Voting of Labels of Nearest Neighbors for Deciding Label of Novice Input

nearest neighbors. The label of the novice item is decided by voting the labels of its nearest neighbors. The ranking selection is adopted for selecting the nearest neighbors from training examples, in this version.

C. System Architecture

This section is concerned with the text summarization system which adopts the table based KNN algorithm. In Section II-B, we described the proposed version of KNN algorithm as the approach to the text summarization. It is viewed as a binary classification which classifies a paragraph into summary or non-summary. This section is intended to describe the text summarization system with respect to its function and its architecture.

The process of sampling paragraphs domain by domain is illustrated in Figure 6. Even if it is possible map the text summarization into an instance of text classification, a paragraph may be classified differently depending on the domain. Sample paragraphs which are labeled with summary or non-summary are gathered domain by domain. The text which is given as the input is tagged with a domain, and the classifier which corresponds to the domain is appointed. The sample paragraphs are encoded into tables in each domain.

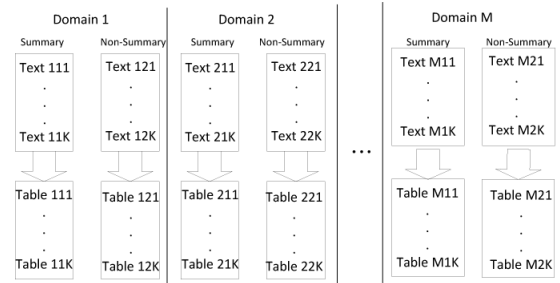


Figure 6. Preparation of Training Examples

The entire architecture of the proposed text summarization system is illustrated in Figure 7. A text is given as the input, and paragraphs are extracted by partitioning the text. In the encoding module, the sample paragraphs in the summary group and the non-summary group and ones which are extracted from the text are mapped into tables in the encoding module. The paragraphs from the text are classified into one of the two categories in the similarity computation module and the voting module. The paragraphs which are classified into summary are extracted as the output in this system.

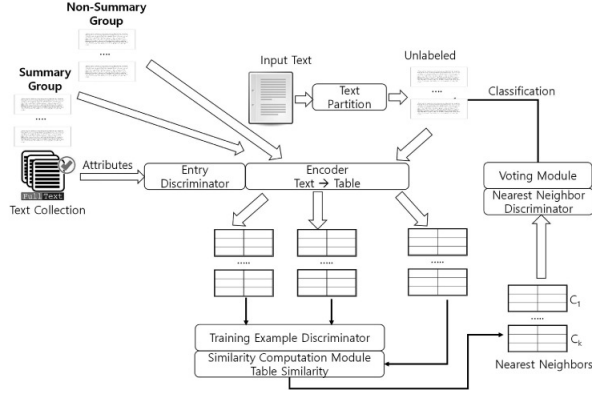


Figure 7. System Architecture

The execution flow of the text summarization system is illustrated in Figure 8. In each domain, sample paragraphs which are labeled with summary or non-summary are gathered. The input text is partitioned into novice paragraphs, and both sample paragraphs and novice paragraphs are encoded into tables. Each paragraph is classified into summary or non-summary, and there are two groups of paragraphs in the text: summary group and non-summary group. The paragraph which are classified into summary are extracted as the summary of the input text in this system.

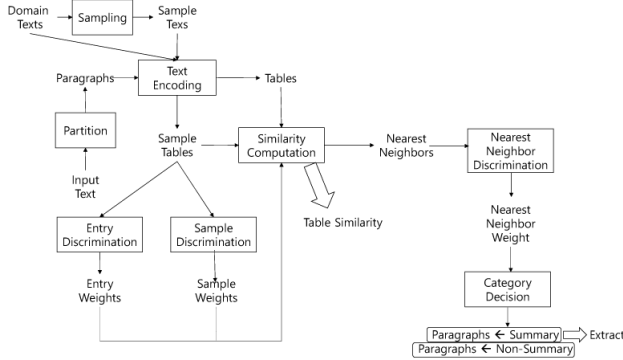


Figure 8. System Flow

Let us make some remarks on the proposed system which is illustrated in Figure 7 as its architecture. The text summarization is defined as the binary classification where each paragraph is classified into summary or non-summary. Paragraphs are encoded into tables instead of numerical vectors, and they are classified directly. Ones which are classified with summary are selected as the summary of the input text. We need to add the text classification module for predicting the domain of input text automatically.

D. Connection with Text Clustering

This section is concerned with the connection of the text summarization with the text clustering. In the previous section, we mentioned the text summarization as an

instance of domain dependent classification. The domains are automatically constructed from the text collection by clustering texts. The AHC algorithm which is proposed in [4] is used for implementing the text clustering. This section is intended to describe the text summarization system which is connected with the text clustering for automating the construction of domains.

The architecture of the text clustering system which was proposed in [4] is illustrated in Figure 9. The texts in the group are encoded into tables, and the AHC algorithm which is proposed in [4] is adopted as the approach. It starts with singletons as many as items, computes the similarities of all possible cluster pairs, and merge the cluster pair with its highest similarity into one cluster. The two steps are iterated until the number of clusters reach the desired number. We may consider replacing the AHC algorithm by its variants for improving the speed and the performance.

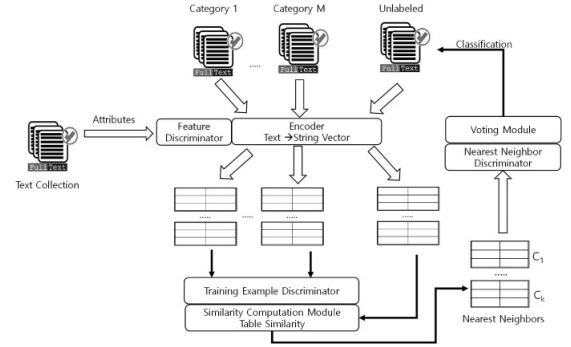


Figure 9. Text Clustering System

The connection of the text summarization with the text clustering is illustrated in Figure 10. The M domains are defined by clustering texts in a group, initially and automatically. A novice text is given as the input, and its domain, domain D , is decided by its similarities with domains. A classifier which corresponds to domain D is nominated and classify each paragraph in a novice text into summary or non-summary. The M text summarization systems are viewed as independent ones, each of which classifies each paragraph into one of the two categories.

The execution flow of text summarization which is connected with the text clustering is illustrated in Figure 11. Unlabeled texts are clustered into subgroups, each of which is a domain. Sample paragraphs which are labeled with summary or non-summary are generated from each domain. These sample paragraphs are provided for training the classifier in each text summarization system. Each classifier is used for judging whether each paragraph in the input text is essential or not.

Let us make some remarks on the connection of the text summarization with the text clustering. It is the process of segmenting a text group into subgroups based on their content-based similarities. The role of text clustering

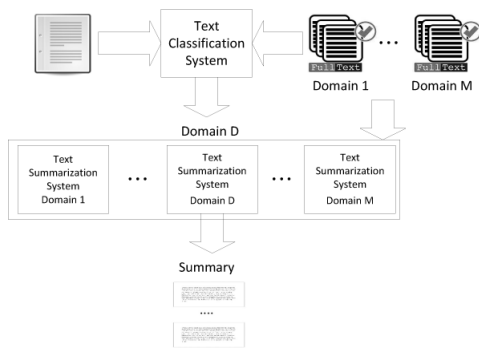


Figure 10. Text Summarization System connected with Text Clustering

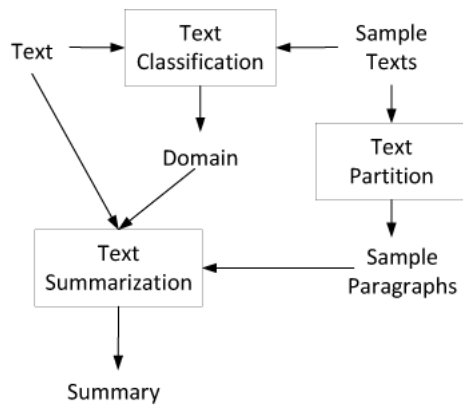


Figure 11. System Flow of Text Summarization connected with Text Clustering

is to define the domains initially in the architecture of connecting the text summarization with the text clustering. In each domain, sample paragraphs are generated, and its corresponding classifier is trained with them. In the future research, we consider using the text summarization for clustering texts in the opposite direction.

REFERENCES

- [1] T. Jo, "Table based Matching Algorithm for Soft Categorization of News Articles in Reuter 21578", 875-882, Journal of Korea Multimedia Society, 11, 2008.
- [2] T. Jo, "Normalized Table Matching Algorithm as Approach to Text Categorization", 839-849, Soft Computing, 19, 2015.
- [3] T. Jo, "Automatic Summarization System in Current Affair Domain by Table based K Nearest Neighbor", 115-121, The Proceedings of 2nd International Conference on Advanced Engineering and ICT-Convergence, 2019.
- [4] T. Jo, "Applying Table based AHC Algorithm to News Article Clustering", 8-11, The Proceedings of International Conference on Green and Human Information Technology, Part I, 2019.